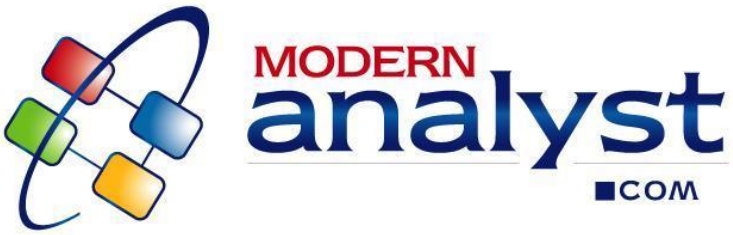

Requirements Document

[Subject/Project]

Revision Date:	00/00/0000
Revision Num:	1.0
File Name:	Requirements Document.doc

Num of Pages:	
Author:	[Author]

Revision History

Date	Revision	Author	Changes
<07/18/2008>	1.0	*[Author]*	*<Original document created>*

Document Reviewers / Approvers

Name	Position	Reviewer	Reviewer and Approver	Signature of Approver	Sign-off Date
<Joe Smith>	*<Business Executive>*		X		
<James Underhill>	*<Analysis Manager>*	X			
<Mary Jane>	*<Development Manager>*	X			

<Note:

- *The text in blue and italic is for guidance only and should be deleted before the document is published.*
- *If any section of this document is not required for the project at hand it may be delete it or its content replaced with "N/A".*
- *All tables and figure in the document will be automatically added to the tables and figures tables of content after they are given a caption.*
- *Go to "File/Properties" and on the "Summary" tab update the following fields to match your project:*
  - *Title – replace the (Title) part of the title with the name of the use case described in this document;*
  - *Subject – enter your project's name here;*
  - *Author – enter your names or first initial followed by your last name;*
  - *Company – enter your company's name.*
- *For the changes from the "Summary" tab to take effect select the all the text in the document and press F9.*
  - *Repeat this in the header and footer areas of the document as well.*
  - *If a message shows up after you've pressed F9, select the "Update entire table" option.*
- *After each update of the document make sure to increase the revision numbers and update the revision date manually on:*
  - *The first page*
  - *The document header*
  - *In the revision table and to describe the changes>*

Table of Contents

Table of Figures

Table of Tables

This page intentionally left blank.

<The document content which starts on the next page should always be on an odd numbered page.>

Requirements Document

1 Executive Summary

<Describe the business problem this project is initiated to solve. Briefly describe the project, including external behavior of the application or subsystem identified, and the functional and nonfunctional requirements.

Specify the purpose and the audience of this document.

This section should not be longer than half a page.

May add a context diagram here.>

2 Assumptions, Restrictions & Limitations

<List any assumptions, restrictions and limitations applicable to the project as a whole, or to some of its parts (specify which one).

Assumptions – Factors or conditions which are considered to be true or to exist without the need to provide documented evidence or empirical data. (per BABOK v.1.6)

Also this section should indicate any design constraints on the system being built. Design constraints represent design decisions that have been mandated and must be adhered to. Examples include software languages, software process requirements, prescribed use of developmental tools, architectural and design constraints, purchased components, class libraries, etc >

3 Data Flow

<The Data Flow diagram shows how information is input, processed, stored and output from a system.

Add a high level data flow diagram with a caption and high level description.

Alternatively may add an activity, workflow or sequence diagram instead. >

4 Customer Workflows

4.1 As-Is Business Use Cases and Actors

The *<figure below>* depicts the 'big picture' of the business processes (business use cases) that will be modified or replaced with the *<Project Name>* project.
The actors involved with each process are also shown.

<Add a business use case diagram>

Figure 41: **Business Use Case Diagram**

4.2 To-Be Use-Cases and Actors

The *<figure below>* shows all use cases and actors involved in the *<Project Name>* project.

- The box represents the scope of this project.
- The ovals inside are the different use cases – sequence of actions that a system performs by interacting with the actors.
- The actors involved with the use cases could be the human beings, other system or pieces of hardware.

<Add a "Actors Generalization and Functional Requirements (Use Cases)" diagram>

Figure 42: **Actors Generalization and Functional Requirements (Use Cases) Diagram**

Actors are user types that represent certain behavioral patterns and are responsible for different operations with different sets of authorities. They represent the customer goals, to be achieved by means of the product. The actor could be a person, another system, a piece of hardware that interacts directly with the system.

Each actor participates in at least one use case or coherent set of use cases to accomplish an overall purpose.

Actor	Background and Skill Set	Goals

Use-cases are collections of scenarios about system use, where each scenario describes a sequence of events. Each sequence is initiated by an actor or by passage of time. The result has to be something of use either to the actor who initiated or to another actor.

Use-Case ID	Actor	Brief Description

5 Business and Functional Requirements

<As this is a new document section, should start on a new page.
In the numbering scheme below, "B" stands for business requirement and "F" for functional requirement. It is designed to allow for requirements traceability when you don't have a requirements management tool.

If preferred the Business and Functional Requirements could be split in two separate sections.>

5.1 General

<Include here only requirements that cannot be added in a separate group or refer to the whole system at high level.>

B 1.1.
- **F 1.1.1.**
- **F 1.1.2.**
- **F 1.1.3.**

B 1.2.
- **F 1.2.1.**

5.2 Section 2

B 2.1.
- **F 2.1.1.**
- **F 2.1.2.**
- **F 2.1.3.**

B 2.2.
- **F 2.2.1.**

5.3 Section 3

B 3.1.
- **F 3.1.1.**
- **F 3.1.2.**
- **F 3.1.3.**

B 3.2.
- **F 3.2.1.**

5.4 Quality of Service Requirements

<If any of the requirements in this section are covered in other documents don't duplicate the content, just reference the other document.

For some projects it may be appropriate to move some of the requirements groups below to a higher level.>

B 4.1. Data Storage

<List what should or should not be stored for historical purposes, in what format and how it is going to be used in the future and when. Do not rely on current defaults as they may change.>

B 4.2. Data De-identification

<List what should or should not be de-identified. Do not rely on current defaults as they may change.>

B 4.3. Data Validation

<List any Data Validation requirements, rules, checks, etc., as applicable for the project.>

B 4.4. Data and User Access

<Identify the user hierarchy, who should be allowed to see and update what type of data. >

B 4.5. Auditing

< How will the user access to the data and the data being modified or used for reporting be tracked, logged and reported.>

B 4.6. Usability

< This section should include all of those requirements that affect usability. For example,
•specify the required training time for a normal users and a power user to become productive at particular operation. For example: The User Interfaces will be designed for usage with minimal or no user training.
•specify measurable task times for typical tasks or base the new system's usability requirements on other systems that the users know and like. For example: The user should be able to reach the widget details after only 3 mouse clicks.
•specify requirement to conform to common usability standards, such as IBM's CUA standards Microsoft's GUI standards>

B 4.7. Reliability

< Requirements for reliability of the system should be specified here. Some suggestions follow:
•Mean Time Between Failures (MTBF) — this is usually specified in hours, but it could also be specified in terms of days, months or years.
•Accuracy—specify precision (resolution) and accuracy (by some known standard) that is required in the system's output.
•Maximum Bugs or Defect Rate—usually expressed in terms of bugs per thousand of lines of code (bugs/KLOC) or bugs per function-point (bugs/function-point).
•Bugs or Defect Rate—categorized in terms of minor, significant, and critical bugs: the requirement(s) must define what is meant by a "critical" bug; for example, complete loss of data or a complete inability to use certain parts of the system's functionality.>

B 4.8. Performance

< The system's performance characteristics should be outlined in this section. Include specific response times. Where applicable, reference related Use Cases by name.

•response time for a transaction (average, maximum)
•throughput, for example, transactions per second
•capacity, for example, the number of customers or transactions the system can accommodate
•degradation modes (what is the acceptable mode of operation when the system has been degraded in some manner)
•resource utilization, such as memory, disk, communications, etc.>

B 4.9. Supportability

< This section indicates any requirements that will enhance the supportability or maintainability of the system being built, including coding standards, naming conventions, class libraries, maintenance access, maintenance utilities.>

B 4.10. System Availability

<Here are some examples:

The reports and software interface created as part of this project will be online / operational during normal business hours.

The maximum allowable downtime will be 3 working days outside the XXXX periods.

During the XXXX periods the maximum allowable downtime will be 8 hours.

System maintenance will be scheduled on a monthly basis and performed from Friday 5:00 p.m. through Monday at 8:00 am. Maintenance at other times can be performed with a minimum of two days advance notice to the affected parties>.

B 4.11. Interfaces

< This section defines the interfaces that must be supported by the application. It should contain adequate specificity, protocols, ports and logical addresses, etc. so that the software can be developed and verified against the interface requirements. For example: - Any software, hardware, data migration or communication interfaces developed as part of this project will be transparent to the user.>

B 4.12. User Interfaces

< Describe the user interfaces that are to be implemented by the software >

B 4.13. Hardware Interfaces

< This section defines any hardware interfaces that are to be supported by the software, including logical structure, physical addresses, expected behavior, etc. >

B 4.14. Software Interfaces

<This section describes software interfaces to other components of the software system. These may be purchased components, components reused from another application or components being developed for subsystems outside of the scope of this document but with which this software application must interact.>

B 4.15. Communication Interfaces

< Describe any communications interfaces to other systems or devices such as local area networks, remote serial devices, etc.>

B 4.16. Data Migration Interfaces

B 4.17. Purchased Components

< This section describes any purchased components to be used with the system, any applicable licensing or usage restrictions, and any associated compatibility and interoperability or interface standards.>

B 4.18. Licensing Requirements

< Defines any licensing enforcement requirements or other usage restriction requirements that are to be exhibited by the software. >

B 4.19. Legal, Copyright and Other Notices

< This section describes any necessary legal disclaimers, warranties, copyright notices, patent notice, word-mark, trademark, or logo compliance issues for the software. >

B 4.20. Applicable Standards

< This section describes by reference any applicable standard and the specific sections of any such standards which apply to the system being described. For example, this could include legal, quality and regulatory standards, industry standards for usability, interoperability, internationalization, operating system compliance, etc. >

F 4.20.1. The ModernAnalyst.com [Company] *<software development>* standards will be applied. *<Specify where these standards could be found – add a link to the document repository.>*

B 4.21. Documentation Requirements

<What user documentation (guides, manuals) will be prepared or updated in relation to the project at hand. Who is responsible for its preparation, review, etc. How this documentation will be distributed to the end users.>

B 4.22. User Acceptance Test Strategy

<Describe the general approach that will be followed during the user acceptance testing of the system, who will perform, report on, facilitate and supervise the UAT. If any UAT documents will be prepared list them here.>

6 Use Cases

<As this is a new document section, it should start on a new page.
There should be one Use Case Section for every system use case.
The diagrams for the main and alternative flows could be the same, with the elements related to the flow at hand highlighted or bolded to stand out. This will provide both an overview of each use case and allow the reader to focus on the flow at hand.
Time permitting, the diagrams steps may be numbered after the use case flow steps in addition to having the same/similar names.>

6.1 Use Case 1

6.1.1 Introduction

Use Case Name	Use Case ID	Date Created	Created By
<Use Case Name>	*<Assign a unique ID (e.g. UC-1)>*	*<Date created (mm/dd/yyyy)>*	*<Analyst's name >*
Primary Actors	**Secondary Actors**		
<Primary actors>	*<Secondary actors>*		

6.1.2 Use Case Description

<Provide a high level description or goal of the use case, typically a couple of sentence. Add the use-case diagram either before or after the description.>

<Add UC 1 Use Case Diagram>

Figure 61: UC 1 – Use Case Diagram

6.1.3 Pre-Conditions

<List anything that must be complete prior to the initiation of the use case. Document these as a numbered list of Pre-Conditions. For example, the user may be required to be logged into the software application or system>.

a)

6.1.4 Trigger

<List the conditions that trigger the use case.>

a)

6.1.5 Post-Conditions

<List the end results expected by the user. Document these as a numbered list of Post-Conditions. For example, the user has saved a template in a designated folder.>

a)

6.1.6 Normal Flow *<Name>*

<Replace <Name> with the normal flow name. The main flow of events covers what "normally" happens when the use case is performed. This is the "Happy day scenario".>

<Add UC 1 Main Flow Activity Diagram>

Figure 62: UC 1 – Main Flow – Activity Diagram

<Add UC 1 Main Flow Sequence Diagram>

Figure 62: UC 1 – Main Flow – Sequence Diagram

<Add UC 1 Main Flow Class Diagram>

Figure 62: UC 1 – Main Flow – Class Diagram

6.1.7 Alternate Flows

6.1.7.1 Alternative Flow <xx> –<Name>

<The alternative flows of events cover behavior of optional or exceptional character in relation to the normal behavior, and also variations of the normal behavior. They handle "what if" scenarios, conditional logic and main flow branching.
Alternative flows = "detours" from the main flow of events
Replace <Name> with the alternative flow's name and <xx> with its number. For alternative flow numbering see the note below.

***Note**: When numbering the alternate flows use the number of the step from which the flow branches and add "A" to indicate that this alternate flow is the first branch from this step, then "B" for the second alternative flow branch from this step, etc. Number the steps in the alternative flow by adding 1,2, etc. after the "A", "B" etc, from the number of the alternate flow.*
Possible alternative flows:

- *Actor selects Yes or No; OK or Cancel, etc.*
- *There is no data*
- *Etc.*

After the UC descriptions add as many diagrams as possible: activity, sequence, flow chart, whatever makes sense. Delete this text before publishing the document.>

<Add UC 1 Alt. Flow <Name> Activity Diagram>

Figure 63: UC 1 – Alt. Flow <Name> – Activity Diagram

<Add UC 1 Alt. Flow <Name> Sequence Diagram>

Figure 63: UC 1 – Alt. Flow <Name> – Sequence Diagram

<Add UC 1 Alt. Flow <Name> Class Diagram>

Figure 63: UC 1 – Alt. Flow <Name> – Class Diagram

6.1.8 Exceptions

6.1.9 Includes

<List any use cases that are included or required for the designated use case.>

6.1.10 Special Requirements

<Describe any special requirements associated with the use case, or their reference number, and the document they are originally described, if not this one.>

6.1.11 Assumptions

<Describe any assumptions associated with the use case, or include their reference number, and location if listed elsewhere.>

6.1.12 Associated Features or Functional Requirements

<Identify associated features or functional requirements, or their reference numbers, for the use case and the document they are originally described.>

<Add UC 1 – State Machine Diagram >

Figure 63: UC 1 – State Machine Diagram

7 Reports

<As this is a new document section, should start on a new page.>

7.1 General

<Include requirements about the lay-out and print-out of the report's header and footer sections that are applicable to all reports listed further down.

If any of the reports has an exception, describe it in that report's section.

Specify the font and size to be used, the horizontal and vertical position and spacing between the elements, when each section should and should not be printed, etc.>

R 1.1. Report Header

R 1.1.1.

R 1.2. Report Footer

R 1.2.1.

R 1.3. Page Header

R 1.3.1.

R 1.4. Page Footer

R 1.4.1.

7.2 Report 1

<This is a list of the requirements needed for correct specification of a report. Their numbering starts with "R" to designate report related requirements.>

R 2.1. *<Brief description>*.

R 2.2. Available for /Report Criteria

R 2.3. Title

R 2.3.1. The report title will be *<Report Title>*.

R 2.3.2. The report description will be *<Report Description>.*

R 2.4. Header and Footer

<List only the exceptions from the requirements in the general section above.>

R 2.5. Special Rules or Conditions

<List any rules for data selection including the report generation criteria chosen by the user and any such criteria that should be applied as default without the user's intervention.>

R 2.5.1.

R 2.6. Fields Order - The report will have the following columns:

<List the data columns (excluding the information from the header and footer sections) that will be on the report in order starting from the top left side of the page. >

Table 71: Report 1 – Fields Order

#	Column Name	Format	Notes
1	*<AAA>*	*<The column content will be centered.>*	
2	*<BBB>*	*<The column content will be left justified.>*	
3	*<CCC>*		*< Using annotation that points to the Notes section of the report.>*

See Figure 71: Report – Sample for more details.

R 2.6.1. *<The Notes section of the report will include the annotation used as field name and the corresponding description of every one of the values displayed>*.

R 2.7. Groups - The data will be grouped by *<report field/criteria>*.

R 2.7.1. Every *<report field/criteria>* will start on a new page.

R 2.7.2. When the next *<report field/criteria>* display starts the current page number will be reset to 1.

R 2.7.3. When the next *<report field/criteria>* display starts the total number of pages number for the report will be reset to correspond to the total pages for the new *<report field/criteria>*.

R 2.8. Sort Order

R 2.8.1. The reports will be produced in ascending order of *<report field/criteria>*.

R 2.8.2. The report will be sorted in ascending order of the *<report field/criteria>*.

R 2.9. Totaling

<Describe which report columns/rows will be totaled, at what level and how often.>

R 2.10. Graph

<Describe the graph's type, position, title, display scale, X and Y axis title, legend, etc..>

R 2.11. Sample

<Add report sample as it should be printed and/or appear on the screen. If the screen and printout looks will be different provide two samples and describe appropriately in the details section above.>

Figure 71: Report – Sample

8 Appendices

<As this is a new document section it should start on a new page.

When numbering requirements in the appendices use the prefix "A" before the number, followed by the appendix number and the sequential number of the requirements in that appendix..

The appendices could include any of the following which have been referenced in the previous document sections::

- *A Create, Read, Update, Delete (CRUD) Matrix.*
- *Class Diagram*
- *Entity Relationship Diagram (ERD)*
- *Data Dictionary – if not created as a separate document*
- *Data Transformation and Mapping tables*
- *Flowcharts*
- *State Machine Diagram – if it spans multiple use cases*
- *User profiles*
- *Etc.>*

8.1 Appendix A

A A.1.

A A.1.1.

A A.2.

A A.2.1.

8.2 Appendix B

A B.1.

A B.1.1.

A B.2.

A B.2.1.

9 Referenced Documents

<For every referenced document provide the following information:

- *Sequence number in the list – the document will be referenced through out this document by that number*
- *Full correct name and full correct location – if the name could also be a link – great.*

Then reference the document number from this section in the document above using "Insert/Reference/Cross-Reference" as this will create a hyperlink from the document text to this list of documents.>